

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ А.С. Савченко
« _____ » _____ 20__ р

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

Тема: «Програмне забезпечення підприємства «Світанок»»

Виконавець: студент _____ УС-411 Гуденко Євген Іванович
(студент, група, прізвище, ім'я, по батькові)

Керівник: к. т. н., доцент _____ Моденов Юрій Борисович
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: ст. викл. _____ Шевченко О.П.
(П.І.Б.) (підпис)

КИЇВ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь **Бакалавр**

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.С. Савченко

“ _____ ” _____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Гуденко Євгена Івановича

(прізвище, ім'я, по батькові)

1. Тема проекту: «Програмне забезпечення підприємства «Світанок»» затверджена наказом ректора № 636/ст від 22.04.2021р.
2. Термін виконання роботи: з 10.05.2021 по 25.05.2021р.
3. Вихідні дані до роботи: Діючі системи автоматизації, інструкції для користувачів, літературні джерела.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, аналітичний огляд, огляд технологій та інструментів для розробки системи, постановка задачі, розробка системи та інтерфейсу для користувачів, висновки.
5. Перелік обов'язкового графічного матеріалу: інформаційна діаграма бази даних системи, схема архітектури додатку.

КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	10.05.2021р. – 12.05.2021р.	
2	Розробка та затвердження плану дипломного проекту.	13.05.2021р.	
3	Аналітичний огляд і постановка задачі.	14.05.2021р.	
4	Порівняльний аналіз існуючих систем.	15.05.2021р. – 16.05.2021р.	
5	Огляд технологій для розробки системи.	17.06.2021р. – 18.05.2021р.	
6	Розробка компонентів системи.	19.05.2021р. – 22.05.2021р.	
7	Висновки та оформлення пояснювальної записки дипломного проекту.	23.05.2021р. – 28.05.2021р.	
8	Підписання необхідних документів у встановленому порядку.	31.05.2021р. – 04.06.2021р.	
9	Підготовка до захисту.	07.06.2021р. – 11.06.2021р.	

Студент

(*Гуденко Є.І.*)

Керівник дипломної роботи

(*Моденов Ю.Б.*)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Інформаційна система управління документами історико-культурної спадщини України»: 56 с., 22 рис., 15 літературних джерел.

Об'єкт дослідження: процес управління готелем.

Предмет дослідження: розробка програми управління готелем.

Мета роботи: проектування та побудова системи управління готелем.

Методи дослідження, технічні та програмні засоби: порівняльний аналіз, розробка бази даних за допомогою DB Browser for SQLite, функціональних компонентів та інтерфейсу системи в середовищі Qt Creator, обробка літературних джерел.

Отримані результати та їх новизна: Отримана програма буде використана для впровадження в готелях з метою підвищення продуктивності та покращення якості обслуговування клієнтів.

АСУ, QT CREATOR, C++, КЛАСИ, ДИНАМІЧНЕ СТВОРЕННЯ

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД І ПОСТАНОВКА ЗАДАЧІ.....	9
1.1. Класифікація готелів	9
1.2. Класифікація готельних номерів.....	14
1.3. Основні служби готелю	16
1.4. Аналіз існуючих систем керування готелем.....	21
1.4.1. Огляд АСУ «Едельвейс»	22
1.4.2. Огляд АСУ «jSolutions».....	23
1.4.3. Огляд АСУ «Fidelio V8».....	24
1.5. Недоліки існуючих АСУ.....	25
1.6. Постановка задачі	25
Висновки до розділу 1.....	26
РОЗДІЛ 2. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТЕМИ УПРАВЛІННЯ	27
2.1. Технології для розробки бази даних.....	27
2.2. Технології для розробки функціональної складової системи.....	29
2.3. Середовище для розробки системи.....	30
Висновки до розділу 2.....	33
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ	35
3.1. Створення бази даних	36
3.2. Створення логіки додатку	38
3.2.1. Створення системи завантаження і запису даних до БД;	38

3.2.2. Створення системи для взаємодії з інтерфейсом.....	39
3.2.3. Створення системи управління.....	42
3.3. Демонстрація роботи програми	43
Висновки до розділу 3.....	46
ВИСНОВКИ.....	47
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ.....	50

ВСТУП

Оптимізація робочого процесу компаній є актуальною на сьогоднішній день. Кожна організація прагне підвищити продуктивність роботи за рахунок впровадження новітніх технологій. З розвитком туризму в нашій країні готельний сервіс став однією з найголовніших галузей у сфері обслуговування. На даний момент у готельному бізнесі багато конкуренції, що підштовхує впроваджувати найсучасніші та найкомфортніші технології в систему керування готелем. Системи керування готелем є дуже складними, адже складаються з багатьох взаємопов'язаних служб та підсистем.

Метою даної роботи є проведення аналізу та створення оптимальної системи управління готелем.

Об'єктом дослідження даної роботи є оптимізація робочого процесу готельного комплексу «Світанок», оскільки кількість персоналу досить велика, і наразі є актуальним створення системи, яка зменшить витрати і підвищить рівень обслуговування клієнтів.

Для досягнення поставленої мети необхідно провести аналітичний огляд поставленої задачі, порівняльний аналіз існуючих систем, дослідити існуючі технології та інструменти для створення системи, розробити базу даних, функціональні компоненти та інтерфейс.

Практичне значення: впровадження розробленої системи управління в робочий процес організації забезпечить виконання наступних функцій:

- надання централізованого місця для зберігання інформації про номери;
- зберігання інформації про клієнтів в базі;
- надання можливості бронювання номерів;
- оптимізація днів та часу заселення;
- оптимізація часу проведення прибирань в номерах.

Розроблена система управління оптимізує робочий процес співробітників готельного комплексу «Світанок» та підвищить ефективність їх роботи, адже вони будуть витрачати меншу кількість часу на пошук необхідної інформації.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД І ПОСТАНОВКА ЗАДАЧІ

У сучасному світі, ми спостерігаємо бурхливий розвиток інформаційних технологій та програмного забезпечення у сфері готельного та туристичного бізнесу. Сучасні інформаційні технології дозволяють організувати роботу компанії з максимальною ефективністю.

Розміщення головний атрибут в комплексі послуг, що надаються туристам під час подорожі. Високий сервіс надання послуг розміщення визначає також такі психологічні аспекти туризму, як задоволеність поїздкою та висока оцінка організації перебування туристів у певному місці [1].

У стандартній класифікації готелів, розробленій експертами Всесвітньої туристичної організації та подібних компаній, вони представляють найбільш комфортну групу колективних засобів проживання для туристів. Зокрема, наступні ознаки:

- надання обов'язкової послуги прибирання кімнат та додаткових послуг пральні, хімчистки, перукарні, прокат автомобілів;
- групування відповідно до вимог національних стандартів на класи і категорії залежно від наданих послуг, наявного обладнання.

1.1. Класифікація готелів

Готелі класифікуються за кількома критеріями:

- рівень комфорту;
- місткість;
- функціональне значення;

Кафедра КІТ				НАУ 21 8 88 000 ПЗ			
Виконав	Гуденко Є.І.			Аналітичний огляд і постановка задачі	Літера	Аркуш	Аркушів
Керівник	Савченко А.С.					9	21
Консульт.					411 122		
Н-контрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

- місце розташування;
- забезпечення харчуванням.

Класифікація готельних компаній за рівнем комфорту відіграє велику роль у вирішенні питань управління якістю готельних послуг.

Критерії рівня комфорту:

- стан і структура кількості кімнат: площа кімнат, частка одномісних номерів, багатокімнатні кімнати, кімнати - квартири;
- стан меблів, інвентарю, санітарно-гігієнічних виробів тощо;
- наявність, стан та режим роботи підприємств харчування: ресторанів, кафе, барів;
- стан будівлі, під'їзні шляхи, облаштування прилеглої до території готелю;
- інформаційна підтримка та технічне обладнання, включаючи наявність телефону, супутникового зв'язку, телевізорів, холодильників, міні-барів, міні-сейфів;
- забезпечити можливість надання ряду додаткових послуг.

Ці параметри оцінюються майже у всіх доступних системах рейтингу готелів. Крім того, ряд вимог пред'являється до персоналу і його підготовці, освіті, кваліфікації, віку, стану здоров'я, знання мов, зовнішньому вигляду та поведінці.

Класифікація готелів по рівню комфортності відповідно з Європейською системою передбачає відповідність кожного класу певній кількості зірок [2]:

- бюджетні готелі «*» – розташовані в центральній частині міста і мають мінімум зручностей;
- готелі туристичного класу «**» – у структурі обов'язкова наявність ресторану і бару;
- готелі середнього класу «***» – рівень обслуговування досить високий;
- готелі першого класу «****» – дуже висока якість комфорту і відмінний рівень обслуговування;
- готелі вищої категорії «*****» – рівень обслуговування і проживання екстра-класу.

Кожен готель має враховувати не тільки рівень комфорту, але також призначення, місце розташування, тривалість перебування клієнта, характеристику архітектури будівлі та деякі інші критерії. Розуміння рівня комфорту як критерію класифікації та процедури присвоєння категорії залежить від рівня комфорту номерів. Вирішити цю проблему також важко, оскільки існують готелі з численними типами розміщення ті зі своєю специфікою [1].

Місткість номерного фонду - це число місць, що може бути запропоновано клієнтам одночасно. Загальноприйнятого підходу до визначення місткості малих, середніх і великих готелів немає. У кожній країні до цього підходять по-різному, з огляду на конкретні особливості соціально-економічного й історичного розвитку.

В Україні готелем вважається заклад, у якому налічується 7 і більше номерів. ЮНВТО рекомендує розуміти під малим готелем, готель з кількістю номерів до 30, одночасно відзначаючи, що це визначення є пріоритетом кожної окремої країни. Для регіону Європи типові невеликі готелі (наприклад, у Німеччині малим вважається готель місткістю до 40 місць, середнім - 40-80 місць, великим - понад 80 місць), а для США і Азії, де швидко розвивається готельний ринок, - готелі великої місткості [2].

Багато країн використовують такий підхід до класифікації готелів за місткістю:

- малі - місткістю до 150 місць (не більше ніж 100 номерів);
- середні - 150-400 місць (до 300 номерів);
- великі - понад 400 місць (понад 300 номерів).

Місткість інших функціональних частин готельного комплексу, зокрема ресторану, кафе, бару, визначається:

- місткістю номерного фонду;
- призначенням готелю;
- наявністю подібних підприємств у прилеглій зоні.

Виходячи з цього критерію, виділяють дві великі групи підприємств:

- транзитні готелі;
- цільові готелі.

Транзитні готелі призначені для обслуговування туристів в умовах короткочасної зупинки. Звичайно вони розміщені уздовж магістралей з великим рухом (залізничних, авіа, авто, водних) [1].

До цільових готелів відносяться готелі ділового призначення і готелі для відпочинку (курортні, туристичні). Готелі ділового призначення обслуговують осіб, що перебувають у ділових поїздках і відрядженнях.

Для обслуговування ділових людей повинні бути створені відповідні умови, у зв'язку з чим до готелів ділового призначення пред'являються такі вимоги:

- місце розташування поблизу адміністративних, громадських та інших центрів ділового життя;
- перевага в номерному фонді одномісних номерів;
- обов'язкова організація в номері, поряд із зоною відпочинку і сну, робочої зони;
- наявність у номерному фонді номерів-апартаментів, необхідних для обслуговування у випадках розміщення ділових людей із членами своїх родин;
- наявність спеціальних приміщень для проведення ділових заходів: конференц-залів (бажано різної місткості), переговорних кімнат, приміщень для експозиції і т. д.;
- наявність спеціального технічного устаткування: для синхронного перекладу, різноманітних видів зв'язку й оргтехніки;
- наявність служб фінансового забезпечення: відділень банків, пунктів обміну валюти і т. д.;
- забезпечення можливості надання високоякісного харчування: наявність ресторанів європейської і національної кухні, кафе і барів вищих категорій, а також доставка харчування в номери;
- устаткування стоянок і гаражів для транспортних засобів і т. д.

У міру розвитку технічного прогресу зростають вимоги клієнтів до послуг, пропонованих готелями ділового призначення. У сучасних умовах ділова людина при виборі готелю звертає увагу на те, щоб у робочій зоні готельного номера поруч

з письмовим столом знаходилася електрична розетка і модемна розетка для підключення до Інтернету й електронної пошти. Телефон, установлений у номері, повинен бути обладнаний кнопкою електронного виклику і мати багато функцій, що дозволить гостю одержувати повідомлення, які прийшли на телефон під час його відсутності в номері.

Концепція "курортного готелю" передбачає надання знаходження, харчування і ряду додаткових послуг людям, що прагнуть до відпочинку і відновлення здоров'я. Найбільш популярним місцем розміщення курортних готелів є території, що надають можливості для відпочинку і лікування в природних кліматичних і природних умовах: на морських узбережжях, у гірських районах тощо [3].

У структурі курортних готелів в обов'язковому порядку повинні бути передбачені приміщення для надання медичних послуг лікувально-профілактичного характеру, надання дієтичного харчування, заняття спортом і активним відпочинком і ін.

Проблеми, з якими зіштовхуються курортні готелі, зумовлені сезонністю попиту, що значною мірою позначається на їх завантаженні. Багато курортів є сезонними по своїй суті (наприклад, зимові чи літні курорти). Тривалість періоду активного функціонування може становити 90-150 днів, що залежить від кліматичних і ряду інших факторів. Щоб вирівняти завантаження в період міжсезоння, останніми роками багато курортних готелів стали прагнути до створення нових ринків, зокрема, усе більша увага цих готелів зосереджується на обслуговування ділового туризму.

Туристичні готелі Сутність концепції "туристичного готелю" зводиться до такого:

- готель, як правило, розташовується на туристичних маршрутах і приймає клієнтів з цих маршрутів;
- комплекс послуг формується відповідно до програми маршруту і заздалегідь визначається ваучером чи путівкою;
- для організації обслуговування в структурі готелю передбачені турбюро або бюро подорожей і екскурсій.

1.2. Класифікація готельних номерів

Готельні номери класифікують за:

- кількістю місць;
- рівнем комфортності;
- призначенням.

У практичній діяльності готельних підприємств набуло поширення розділення номерів в залежності від виду компонування ліжок [2]:

- одномісні номери (SGL);
- одномісний номер для 1 дорослого з дитиною (SGL + CHD);
- одномісний номер для 1 дорослого з 2 дітьми (SGL + 2 CHD);
- двомісні номери (DBL);
- двомісний номер з додатковим ліжком (DBL + EXB);
- двомісний номер для двох дорослих з дитиною до 12 років (DBL + CHD);
- двомісний номер для двох дорослих і двох дітей (DBL + 2 CHD);
- двомісний номер для трьох дорослих з додатковим ліжком і дитиною (DBL + EXB + CHD).
- тримісні номери (TRPL).
- тримісний номер для трьох дорослих з дитиною (TRPL + 1 CHD);
- тримісний номер для трьох дорослих з двома дітьми (TRPL + 2 CHLD).

Використання здвоєних стандартних ліжок (DBL) дозволяє перетворювати номер в одномісний або двомісний в залежності від попиту. Використання ліжок дуже великого, «королівського розміру» в обладнанні номерів говорить про підвищений рівень комфорту, оскільки розміри таких ліжок перевищують стандартні, номер повинен мати чималу площу.

Класифікація готельних номерів по рівню комфортності [2]:

- Номери поліпшеного планування категорії напівлюкс (Junior Suite);
- Номери поліпшеного планування, переважно складаються з 2 кімнат – відносяться до категорії люкс (Suite);
- Номери підвищеної комфортності (De Luxe);

- Дворівневі номери – дуплекси (Duplex);
- Сімейні двокімнатні і трикімнатні номери (Family Room);
- Однокімнатні номери категорії студія с міні-кухнею (Studio);
- Стандартні однокімнатні номери (Standart);
- Бунгало – окремі будиночки-бунгало. (Residence, Bungalow, Village, Chalet, Cabana);
- Окремі апартаменти з власною кухнею (Apartament);
- Номери для молодят – відносяться до категорії люкс (Honeymoon Room);
- Вілли – малі, люкс, двоповерхові. Часто мають власний сад і басейн. (Villa, Villa Deluxe, Junior Villa. President Villa).

Класифікація готельних номерів по призначенню:

- бізнес-класу;
- економ-класу;
- номери-апартаменти.

Номери бізнесу-класу призначені для розміщення осіб, що знаходяться в ділових поїздках і відрядженнях. Крім стандартної обстановки номера, тут повинні бути створені умови для роботи, а саме: повинен бути робочий стіл, телефон, факс, комп'ютер і т. д [1].

Номери економічного класу призначені для розміщення широкого контингенту клієнтів. Відзначаються недорогим інтер'єром і устаткуванням, що впливає на ціну розміщення.

Номери-апартаменти - 2-, 3-, 4-кімнатні номери призначені, як правило, для довгострокового розміщення родин. У складі кімнат номера-апартаменту обов'язково повинна бути кухня з необхідним набором побутової техніки (кавоваркою, мікрохвильовою піччю, міксером і т. д.), що дозволяє забезпечити практично домашні умови перебування.

Розрізняють також таку категорію номерів, як сюїт-номери – 3, 4-кімнатні номери з житловою площею не менше ніж 45 м², у складі яких кухні може й не бути.

1.3. Основні служби готелю

Для здійснення замкнутого технологічного циклу обслуговування гостей у готелі передбачені основні служби:

Першою є служби бронювання, яка приймає та оброблює заявки, а також складає необхідну документацію[3].

Заявки приймають по телефону, факсу, поштою, за допомогою комп'ютерних систем бронювання. Кожна заявка містить таку інформацію:

- дату і час заїзду;
- дату і час від'їзду;
- кількість гостей;
- категорію номера;
- послуги в номері;
- послуги харчування;
- ціну;
- прізвище того, хто буде оплачувати рахунок (або назва фірми);
- вид оплати (нааявний, безготівковий, з використанням кредитної картки);
- особливі побажання (заздалегідь забронювати стіл у ресторані, трансфер, можливість тримати в номері тварин і т. д.).

Після відповідного оброблення заявки співробітником служби бронювання гостю направляють підтвердження або відмовлення.

Підтвердження заявки - це спеціальне повідомлення про те, що гостю буде надане розміщення в готелі. Звичайно на повідомленні вказують номер підтвердження, дату передбачуваного прибуття і вибуття гостя, категорію замовленого номера, кількість гостей, кількість ліжок та інші вимоги, що спеціально обумовлені. Для того щоб ще раз уточнити всі деталі розміщення, а також виключити виникнення спірних питань, бажано, щоб після прибуття в готель повідомлення було в гостя із собою [1].

У своїй діяльності готельні підприємства дуже часто звертаються до гарантованого підтвердження заявок. Це означає, що вони підтверджують

бронювання тільки після одержання від клієнта відповідних гарантій оплати на випадок, якщо клієнт прибуде із запізненням чи взагалі не прибуде. Такими гарантіями насамперед є передоплата в розмірі 50 % чи 100 % вартості добового розміщення або розміщення протягом усього терміну, а також інформація про номер кредитної картки клієнта. Проте в цьому випадку необхідно, щоб у законодавстві держави, в якій розташований готель, містилися строгі норми відповідальності за відмову клієнта від підтвердженого готелем бронювання.

З погляду гостей, служба обслуговування є найважливішою в готелі, тому що персонал саме цієї служби працює з клієнтами в постійному контакті і виконує всі функції, пов'язані з їхнім безпосереднім обслуговуванням.

Службу обслуговування очолює менеджер, якому підлегли швейцари, коридорні, підношувачі багажу, ліфтери, консьєржі, розсильні, водії [4].

Перші враження клієнтів про готель надто важливі, тому на персонал служби обслуговування покладається особлива відповідальність. Як правило, першими зустрічають гостей швейцари, що стоять біля входу в готель. Вони повинні привітати гостей, допомогти їм вийти з машини. Швейцари повинні володіти інформацією про послуги, що маютьсЯ в готелі, про готельні заходи, про місце розташування готелю.

Супроводять гостей у номери, а також доставляють багаж коридорні. Під час супроводу їм рекомендується підтримувати з гістьми розмову. При цьому особливо важливо дати інформацію про існуючі в готелі послуги: наявність і режим роботи ресторану, кафе, бару, пральні, хімчистки, басейну, спортзалів і т. д.

Після прибуття в номер коридорний повинен допомогти гостю розміститися: пояснити (і одночасно перевірити справність), що і як працює.

Безліч важливих послуг роблять гостям консьєржі. До найбільш типових послуг, що виконуються консьєржами, відносяться [3]:

- придбання і доставка театральних квитків;
- замовлення столика в міських ресторанах;
- замовлення і доставка авіа-, залізничних, автобусних квитків, довідки про роботу міжнародного, міжміського і внутрішнього транспорту;

- резервування місць у перукарню, косметичний салон, на прийом до лікаря;
- інформація про місцеві визначні пам'ятки, роботу музеїв, виставок, магазинів;
- допомога в екстрених випадках;
- виконання особистих доручень клієнтів.

Для того щоб якісно виконувати всі ці послуги, консьєржам необхідно в усіх деталях знати життя готелю в цілому і його численних служб зокрема, говорити кількома іноземними мовами, бути комунікабельними, при будь-яких обставинах демонструвати дипломатичний такт і зберігати прекрасний настрій.

Службу прийому важлива частина готелю. З цією службою гість контактує найбільше, сюди ж він звертається за інформацією і послугами під час свого перебування в готелі [3].

До найважливіших функцій служби прийому відноситься вітання гостей і виконання необхідних формальностей при його розміщенні. Співробітник служби прийому (порт'є) після швейцара, що стоїть біля входу, практично є першим працівником готелю, з яким контактує гість. Від того, як приймуть гостя, як його привітають, як швидко будуть виконані необхідні формальності (перевірка бронювання, заповнення анкети, передоплата), багато в чому залежить перше враження від готелю в цілому.

При розміщенні слід ще раз погодити умови попереднього бронювання (категорію номера, наявність зручностей, вид оплати, ціну, термін перебування, передбачувану дату від'їзду і т. д.). Якщо гостю необхідно заповнити анкету, варто зробити цю процедуру швидко. Наприклад, якщо гість уже бував у цьому готелі, тоді досить тільки його підпису на анкеті (мається на увазі, що інформація про гостя збережена з часу його першого перебування в готелі).

До функцій служби прийому і розрахункової частини відносяться також розподіл номерів і облік вільних місць у готелі, виписування рахунків і розрахунки з клієнтами.

До прибуття гостей служба прийому одержує зі служби бронювання оброблені заявки, відповідно до яких складає карту руху номерного фонду, що

допомагає вести облік вільних місць у готелі. У більшості великих готелів облік вільних місць робиться автоматично з використанням спеціальної комп'ютерної програми. У малих і середніх готелях донедавна (в окремих готелях - і зараз) для цього використовувався спеціальний номерний стенд, що має осередки, куди вставлялися різнобарвні картки відповідно до кількості днів перебування клієнта в готелі. Різні кольори дозволяють бачити різні категорії гостей. Наприклад, зелений колір використовується для позначення групових туристів, жовтий - учасників конгресу, червоний - індивідуальних, білий - ділових і т. д. Такий номерний стенд дає наочну картину завантаження номерного фонду і наявності вільних місць. Зведення про наявність вільних місць у готелі й кількості проживаючих гостей чітко й оперативно передаються службою прийому всім іншим службам готелю.

Для збереження ключів від номерів використовується стенд із пронумерованими відділеннями. Історично ці стенди розміщували так, щоб їх було видно і співробітникам готелю, і гостям. Таке розміщення дозволяло всім за наявності ключів судити про те, зайнятий номер чи ні. Зараз підхід до організації таких стендів змінюється: найчастіше їх розміщують у схованих відділеннях стійки для забезпечення безпеки і нерозголошення інформації про зайнятість номера [3].

Відразу ж після розміщення гостей у готелі розрахункова частина відкриває на його ім'я рахунок. Звичайно гість користується послугами різних служб готелю, пред'являючи візитну картку або називаючи номер кімнати. Ці служби подають зведення про витрати гостей в розрахункову частину, що включає їх у загальний рахунок. Наприкінці перебування гостей розрахункова частина розраховується з ним та інформує про це всі готельні служби.

Одна з функцій служби прийому- ведення картотеки гостей. На кожного гостя після його перебування в готелі заповнюють спеціальну картку, в якій міститься інформація, зібрана з усіх служб готелю, що мали з гостем контакт. При повторному приїзді гостей ця інформація дає можливість персоналу значно підвищити якість обслуговування, дозволяє багато в чому передбачити переваги і побажання гостя. Подібна робота дуже важлива для цілей придбання постійних клієнтів і є

додатковою рекламою готелю. На сучасних готельних підприємствах установлені комп'ютерні програми дають змогу створити великий банк даних про клієнтів.

У деяких закордонних готелях зі сформованими традиціями обслуговування в складі служби прийому і розрахункової частини є посада нічного портьє. Він наділений широким колом функцій і обов'язків, що в два рази перевищують обов'язки звичайного портьє. Нічний портьє, як правило, змінює денного не раніше ніж о 23.00, приймаючи на себе насамперед функції прийому і розрахункової частини. Після 24.00 нічний портьє часто виконує функції бармена вестибюльного бару, що працює цілодобово.

Однією з основних обов'язків нічного портьє є обхід готелю, щоб перевірити справність сигналізації (охоронної і протипожежної) і безпеку готелю. Під час обходу нічний портьє збирає аркуші-замовлення сніданків у номерах, які заповнюють клієнти і вивішують за двері. У нічний час таких обходів відбувається не менше трьох. Час кожного заноситься в спеціальний журнал.

Після того як зібрані всі аркуші-замовлення сніданків у номер, нічний портьє вводить у журнал інформацію про вид сніданків (звичайно "континентальний" або "континентальний плюс"), їхню вартість і кількість для кожного номера.

Найважливішою функцією служби експлуатації номерного фонду є підтримка необхідного рівня комфорту та санітарно-гігієнічного стану готельних номерів, а також суспільних приміщень. За чисельністю зайнятого персоналу ця служба найбільша в готелі. Як правило, тут працює до 50 % усіх службовців готелю.

Службу експлуатації номерів очолює менеджер, якому підлегли покоївки, чергові на поверхах, супервайзери, стюарди та деякі інші категорії працівників.

Основним обов'язком покоївок є прибирання номерів, незалежно від того, зайняті вони чи вільні. Прибирання номерів буває щоденним, після виїзду проживаючого та генеральним. Щодня покоївка виконує поточне і проміжне збирання номерів.

Прибирання номерного фонду здійснюється в такій послідовності: спочатку роботи ведуться в заброньованих номерах, потім прибираються номери, які звільнилися від проживаючих гостей, в останню чергу виконуються прибирання в

зайнятих приміщеннях. Прибирання варто робити під час відсутності гостя. Якщо ж гість знаходиться в номері, необхідно одержати в нього дозвіл на прибирання.

Служба безпеки готелю забезпечує безпеку гостей та їхньої власності. Цей фактор усе частіше враховується клієнтами при виборі готелю.

До вирішення питань охорони та безпеки готельні підприємства підходять по-різному. Деякі невеликі підприємства надають право управління безпекою сторонній організації (приватним охоронним агентствам, міліції). Великі готелі, як правило, у своїй структурі створюють власну службу безпеки і включають відповідних фахівців у штат співробітників. У світовій практиці для визначення кількісного та якісного складу персоналу служби безпеки вважається раціональним, з погляду рівня витрат і ступеня забезпечення безпеки, правило "300", тобто один співробітник служби безпеки на 300 номерів. При цьому ґрунтуються на ємності, а не на завантаженні номерів: якщо в готелі є 300 номерів і більше, то необхідна наявність штатного співробітника служби безпеки.

1.4. Аналіз існуючих систем керування готелем

Готельний бізнес у нашій країні є провідною галуззю сфери послуг, яка з кожним роком стає ще більш актуальною. Бурхливий розвиток туризму не міг не спонукати власників готельних комплексів впроваджувати сучасні технології, тим самим покращувати якість і швидкість обслуговування. Однією з таких інновацій є автоматизована система управління готелем.

На ринку представлено більше десятка пропозицій, за допомогою яких може проводитися автоматизація готелю. Серед досить відомих можна відзначити:

- SmartHotel;
- Fidelio V8;
- Едельвейс;
- 1С: Готель.
- jSolutions

Вони відрізняються між собою за технічними параметрами, ступеня надійності, функціональності. Розробки в цьому напрямку тривають, що свідчить про недосконалість вже випущених програмних рішень.

1.4.1. Огляд АСУ «Едельвейс»

Бронювання і скасування номерів відбувається візуально в інтерфейсі, що дозволяє робити цю операції швидко, зручно і просто. Також спокійно відстежується весь процес задля контролю. Інтерфейс є дуже простим та інтуїтивно зрозумілим навіть недосвідченим користувачам. Це надає змогу будь-якому клієнту користуватися системою [5].

На схемі (рис. 1.1) показані основні блоки структури готелю, які використовуються системою «Едельвейс». Система побудована в архітектурі «клієнт-сервер».

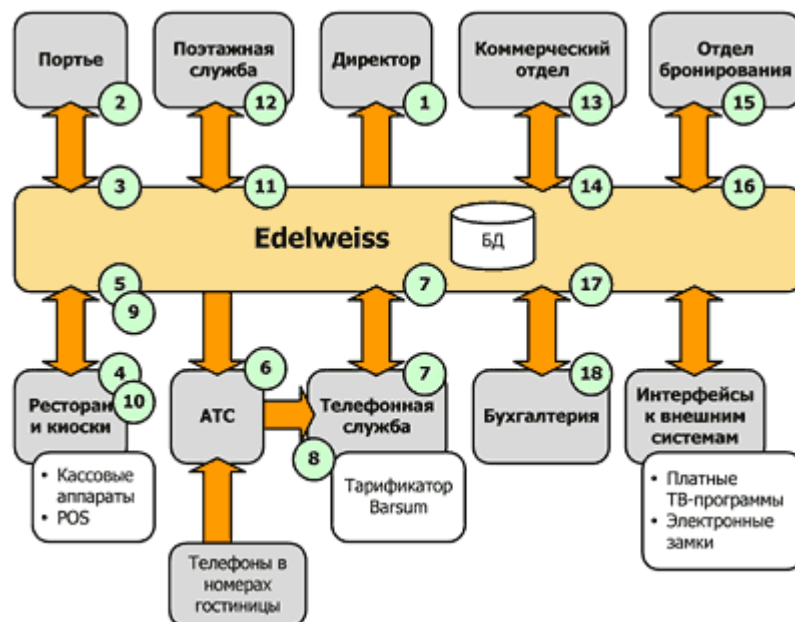


Рис. 1.1. Структура АСУ «Едельвейс»

Недоліки: висока вартість програми (оплата відбувається щомісячно), яка складається з кількості ліцензій, кількості модулів, а також набору інтерфейсів для взаємодії з суміжними системами.

1.4.2. Огляд АСУ «jSolutions»

«jSolutions» є хмарною системою для автоматизації управлінських та облікових завдань підприємств. jSolutions дозволяє повністю автоматизувати бізнес-процеси та мінімізувати витрати на користування системою. Низька загальна вартість володіння програмним забезпеченням. Можливість роботи системи на будь-якій операційній системі. Система інтегрується з різноманітним обладнанням, та може працювати на різноманітному устаткуванні [6].

Комп'ютерна програма для управління готельним бізнесом jSolutions дозволяє вести комплексне управління діяльністю закладу, володіти оперативною інформацією, уникати впливу людського фактора.

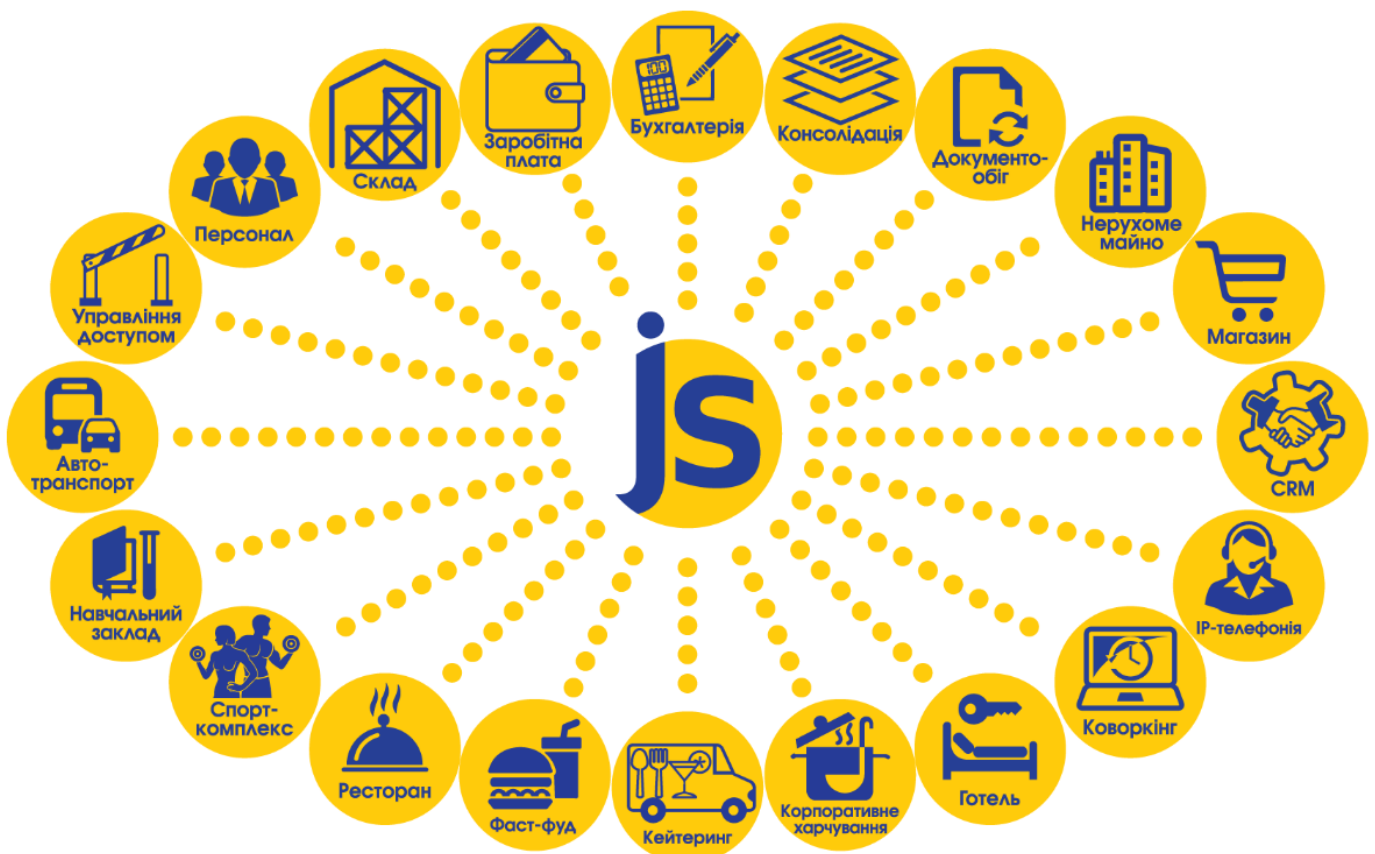


Рис. 1.2. Можливості АСУ «jSolutions»

Недоліки: помісячна оплата програми, необхідність в проходженні тренінгів, щоб користуватись функціоналом.

1.4.3. Огляд АСУ «Fidelio V8»

Fidelio V8 – це єдина база даних під управлінням однієї із самих надійних та гнучких у світі систем управління базами даних (СУБД) Oracle. Тепер усе: від здачі номера в готелі та управління зв'язками з клієнтами до звітності для управління та контролю за діяльністю готелю – об'єднані в єдину систему, у центрі якої знаходиться гість [7].

Fidelio інтегрована з найбагатшим набором використовуваних в готелі систем: ресторанної, замкової, телекомунікаційної, платного телебачення, автоматичних мінібарів, авторизації кредитних карт та енергозбереження.



Рис. 1.3. Складові АСУ «Fidelio V8»

Недоліки: відсутня українська локалізація, інтерфейс заповнений не потрібною інформацією, що ускладнює користування.

1.5. Недоліки існуючих АСУ

Незважаючи на те, що впровадження автоматичної системи управління має багато переваг, були виявлені деякі критичні недоліки, такі як [8]:

- Впроваджені в системі технології можуть бути занадто модифікованими та складними, що збільшить час впровадження та вивчення принципів роботи;
- Високі витрати на впровадження та підтримку програмного забезпечення;
- Типові операції можуть бути реалізовані як окремі компоненти системи, тобто, можлива відсутність єдиного інтерфейсу для забезпечення всіх наданих функцій.

У зв'язку з появою можливості бронювання номерів онлайн, організації розпочали захищати данні додатково, що зменшує швидкодію системи в цілому.

Також організації при впровадженні АСУ можна перевантажити інформацією, що може спричинити втрату контролю, або погіршення в продуктивності роботи компанії.

1.6. Постановка задачі

Завдання полягає у створенні застосунку для менеджменту номерів. Для створення застосунку були приведені такі вимоги як:

- українська локалізація;
- зручний і простий інтерфейс;
- можливість використання застосунку на Windows та Linux;
- мінімальне використання ресурсів.

Програма повинна виконувати такі функції:

- можливість бронювання номерів;
- сортування номерів за необхідними параметрами
- автоматична оптимізація заселення;
- ведення бази даних клієнтів;

Спроектована система управління дозволить оптимізувати робочий процес співробітників готельного комплексу «Світанок» та підвищить ефективність їх роботи, адже вони будуть витрачати меншу кількість часу на пошук даних та приділяти основну увагу роботі з клієнтами.

Висновки до розділу 1

Для готелів питання комплексної автоматизації процесів бронювання місць, розрахунків з гостями і проблеми безпеки стають все більш актуальними. Враховуючи всю зібрану інформацію, що велика кількість створених автоматизованих систем управління мають наступні ключові недоліки: надмірна функціональність, необхідність щомісячної оплати за використання сервісів, надмірна складність освоєння. Отже, виникла необхідність створення власної системи для уникнення зазначених проблем

РОЗДІЛ 2

ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТЕМИ УПРАВЛІННЯ

В додатку визначаються рівні призначені для інтерфейсу користувача, бізнес-логіки і доступу до даних. В рамках такої архітектури користувачі виконують запити через інтерфейс користувача, який взаємодіє тільки з рівнем бізнес-логіки. Рівень бізнес-логіки, в свою чергу, може викликати шар доступу до даних.

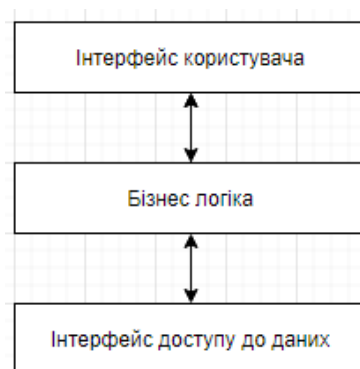


Рисунок 2.1. Архітектура додатка

2.1. Технології для розробки бази даних

Існує два основних типи баз даних: реляційні та нереляційні, основна відмінність між якими полягає в методах зберігання і обробки даних [9].

Реляційні бази даних зберігають інформацію у вигляді логічних структур – таблиць, які можуть бути пов’язані між собою. Кожен рядок таблиці – це запис із унікальним ідентифікатором. Стовпці таблиці містять атрибути даних, на основі яких будуються первинні та зовнішні ключі для можливості встановлення зв’язків.

Кафедра КІТ (47)				НАУ 21 8 88 000 ПЗ			
Виконав	Гуденко Є.І.			Технології та інструменти для створення системи управління	Літера	Аркуш	Аркушів
Керівник	Моденов Ю.Б.					27	7
Консульт.							
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						
					411 122		

На сьогодні більшість сховищ інформації для додатків розроблено за допомогою реляційних систем управління базами даних, найпопулярнішими з яких є: MySQL, PostgreSQL, Db2.

Нереляційні системи управління базами даних є більш гнучкими та адаптивними і використовуються при розробці сховищ для додатків, які обробляють велику кількість об'ємних даних, між якими складно визначити взаємозв'язки.

SQL - декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

Перед початком розробки бази даних за допомогою SQL необхідно провести інформаційне моделювання, яке полягає у виконанні наступних етапів:

- 1) Визначення основних сутностей та встановлення зв'язків між ними;
- 2) Визначення атрибутів сутностей;
- 3) Побудова E-R діаграми;
- 4) Перетворення E-R діаграми в реляційну конструкцію;
- 5) Визначення первинних та зовнішніх ключів для сутностей.

Сутності на E-R діаграмі, представленої на рис. 2.2, зображуються прямокутниками, а зв'язки – лініями, що з'єднують їх. Існує три типи зв'язків між сутностями: один до одного, один до багатьох, багато до багатьох. Для зображення невизначеності зв'язку використовується коло, для зображення того, що зв'язок тільки з одним примірником – риска, а для зображення зв'язку з багатьма примірниками – трикутник [9].

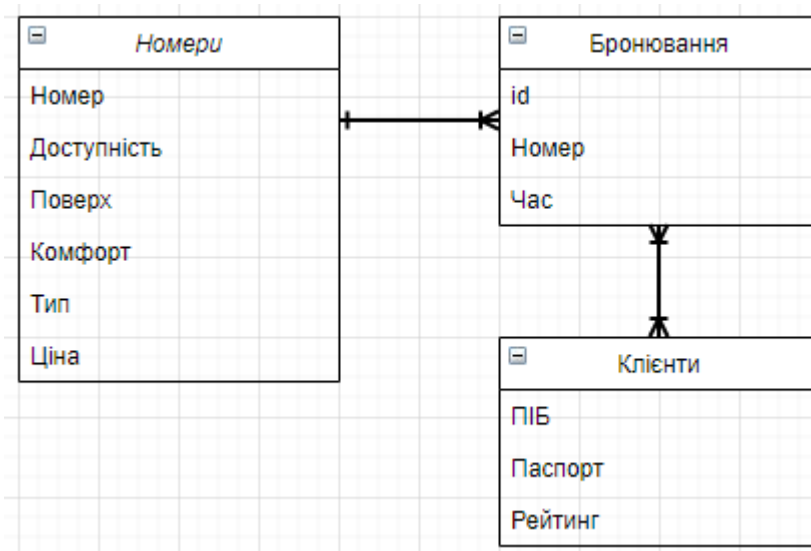


Рис. 2.2. Інформаційна діаграма «сутність-зв'язок» (E-R) бази даних

Для перетворення E-R діаграми в реляційну модель необхідно визначити первинні та зовнішні ключі у сутностей та описати типи даних атрибутів. Для сутності «номери» первинним ключем є номер, оскільки він є унікальним. Оскільки всі сутності бронювання мають бути пов'язані з об'єктом, до кожної з них додається зовнішній ключ – номер, який буде містити посилання на необхідний об'єкт.

Результатом виконання зазначених етапів є сформована інформаційна модель, яка є основою для розробки бази даних за допомогою SQL.

2.2. Технології для розробки функціональної складової системи

Програма написана на мові C++. C++ - мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної.

В C++ немає типів даних і елементарних операцій високого рівня. Наприклад, не існує типу матриця з операцією звернення або типу рядок з операцією конкатенації. Якщо користувачеві знадобляться подібні типи, він може визначити їх в самій мові. Програмування на C++ по суті зводиться до визначення універсальних або залежних від галузі застосування типів. Добре продуманий користувацький

тип відрізняється від вбудованого типу лише способом визначення, але не способом застосування [10].

З мови виключалися можливості, які можуть привести до накладних витрат пам'яті або часу виконання, навіть якщо вони безпосередньо не використовуються в програмі. Наприклад, було відкинуто пропозицію зберігати в кожному об'єкті деяку службову інформацію. Якщо користувач описав структуру, яка містить дві величини, що займають по 16 розрядів, то гарантується, що вона поміститься в 32-х розрядний регістр.

Мова C++ проектувався для використання в досить традиційному середовищі, а саме: в системі програмування C операційної системи UNIX. Але є цілком обґрунтовані аргументи на користь використання C++ в багатшою програмному середовищі. Такі можливості, як динамічне завантаження, розвинені системи трансляції та бази даних для зберігання визначень типів, можна успішно використовувати без шкоди для мови.

Типи C++ і механізми приховування даних розраховані на певний синтаксичний аналіз, проведений транслятором для виявлення випадкового псування даних. Вони не забезпечують секретності даних і захисту від навмисного порушення правил доступу до них. Однак, ці кошти можна вільно використовувати, не боячись накладних витрат пам'яті і часу виконання програми.

2.3. Середовище для розробки системи

Інтегроване середовище розробки або IDE - це додаток або програмне забезпечення, яке програмісти використовують для створення програмного продукту. IDE допомагає програмісту легко програмувати, надаючи всі комплексні засоби, необхідні для розробки програмного забезпечення. IDE може підвищити продуктивність програміста або розробника завдяки швидкому налаштуванню і різним інструментам. Без цього програмісту потрібно багато часу, щоб вибрати різні інструменти для вирішення своїх завдань.

В основному IDE складається з трьох частин: редактора вихідного коду, засоби автоматизації збирання (компілятора) і відладчика. Редактор вихідного коду - це те, де програмісти можуть писати код, тоді як інструмент автоматизації збирання використовується програмістами для компіляції коду, а відладчик використовується для тестування або налагодження програми для усунення будь-яких помилок в коді. Крім того, IDE також мають додаткові функції, такі як моделювання об'єктів і даних, модульне тестування, бібліотеку вихідного коду і багато іншого.

На сьогоднішній день найпопулярнішими IDE, які використовуються для розробки програмного забезпечення на C++, є Visual Studio, Visual Studio Code і MonoDevelop, Qt Creator.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ [12].

Visual Studio Code - відмінний високопродуктивний легкий редактор, який пропонує хорошу підтримку завершення проекту. Працює на Node JS [13].

MonoDevelop підходить для швидкої розробки настільних і web-додатків. IDE дозволяє переносити додатки, написані в Visual Studio, на Linux і Mac OS X, оскільки підтримує єдину базу коду платформ [14].

Для розробки програми в даній роботі було обрано Qt Creator. Починаючи з версії 4.5.0 в комплект Qt включена середовище розробки «Qt Creator», яка включає в себе редактор коду, довідку, графічні засоби «Qt Designer» і можливість налагодження додатків. «Qt Creator» може використовувати GCC або Microsoft VC++ як компілятора і GDB як відладчика. Для Windows версій бібліотека 10 комплектується компілятором, заголовними і об'єктними файлами MinGW [11].

Qt Creator - кроссплатформенная вільна IDE для розробки на C, C ++ і QML. Розроблено Trolltech (Digia) для роботи з фреймворком Qt. Основне завдання Qt Creator - спростити розробку програми на різних платформах. Тому серед

можливостей, властивих будь-якому середовищі розробки, є і специфічні, такі як налагодження додатків на C++ і відображення в відладчику даних з контейнерів Qt, вбудований дизайнер інтерфейсів як на QML, так і на Qt Widgets. Qt Creator підтримує системи збирання qmake, cmake, autotools. Для проектів, створених під іншими системами, може використовуватися в якості редактора вихідних кодів. Є можливість редагування етапів збірки проекту. В Qt Creator реалізовано автодоповнення, в тому числі ключових слів, введених в стандарті C++11, підсвічування коду. Також є можливість завдання стилю вирівнювання, відступів і постановки дужок. Реалізовано ряд можливостей при роботі з сигнатурами методів, а саме: автогенерація порожнього тіла методу після його поновлення; можливість автоматично змінити сигнатуру методу у визначенні, якщо вона була змінена в оголошенні і навпаки; можливість автоматично поміняти порядок проходження аргументів.

При навігації по коду є перемикання між визначенням і оголошенням методу, перейти до оголошення методу, перейменування методу як в окремому проекті, так і в усіх відкритих. Також є можливість викликати довідку згідно з поточним контексту.

Побудова інтерфейсу даної програми буде відбуватися на Graphics View Framework. Основу Graphics View Framework складають три класи Qt Library. Клас QGraphicsScene надає поверхню для управління великим числом графічних 2D елементів. Графічна сцена (QGraphicsScene) зберігає в собі графічні елементи (QGraphicsItem). Елементи упорядковуються певним чином, за рахунок цього, забезпечуються оновлення мінімальної області сцени при зміні властивостей елемента і ефективно виявлення зіткнень.

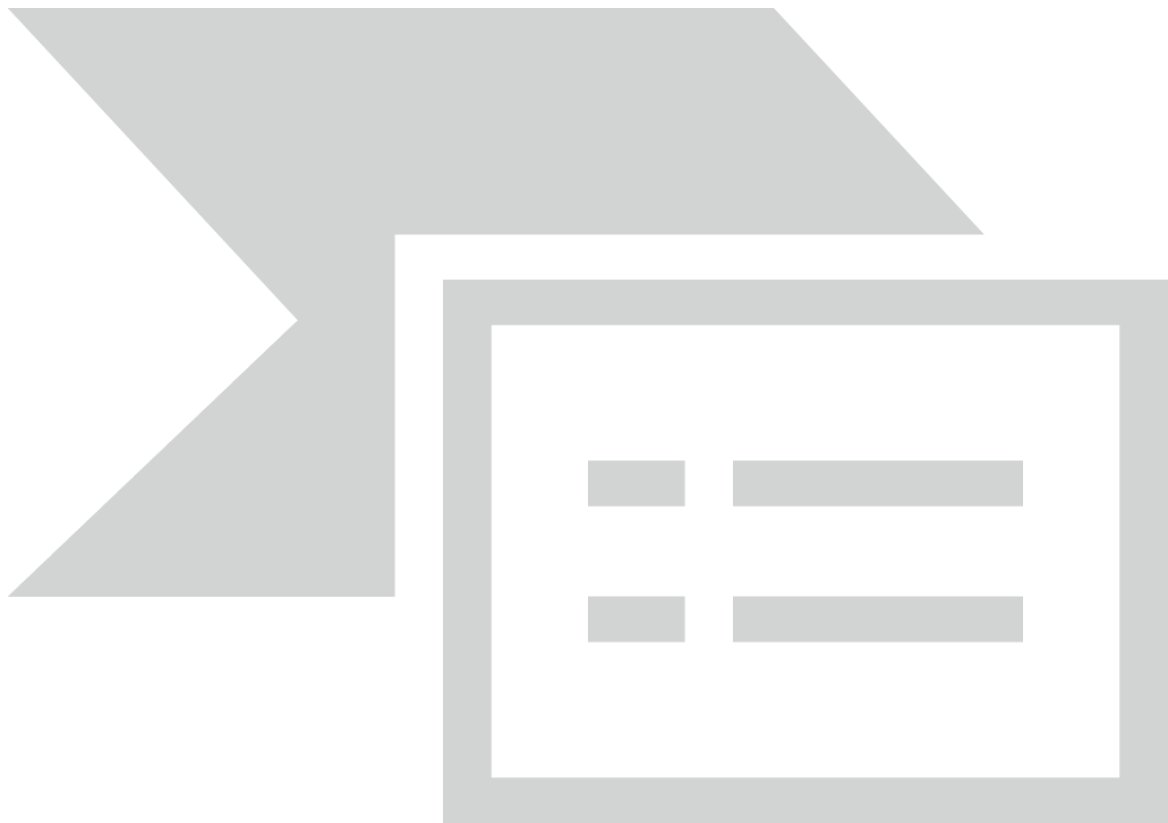


Рис. 2.3. Схема Graphics View Framework

Висновки до розділу 2

Для розробки автоматичної системи управління готелем було використано декілька технологій, які об'єднані в єдиному IDE Qt Creator.

Для розробки програми було обрано об'єктно-орієнтовану мову програмування C++. Програмування на C++ зводиться до визначення універсальних або залежних типів даних. Добре створений тип забезпечує відсутність витрат пам'яті або часу виконання. Основними перевагами такого методу програмування є оптимальне використання пам'яті, кросплатформеність, стабільність та велика бібліотека вбудованих класів.

Для розробки інтерфейсу користувача було використано Graphics View Framework, який представляє собою три класи Qt Library, які забезпечують виведення графіки на екран.

Безпосередньо для створення компонентів системи було обрано інтегроване середовище розробки Qt Creator – середовище для написання, відладки і запуску коду.

Інтегроване середовище розробки являє собою багатофункціональну програму, яку можна використовувати для розробки програмного забезпечення. Також Qt Creator включає в себе компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для спрощення процесу розробки.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМИ

Файлова структура проекту (рис. 3.1), складається з директорій, в якій знаходяться ключові компоненти, кожний з яких відповідає за окремий функціональний компонент системи.

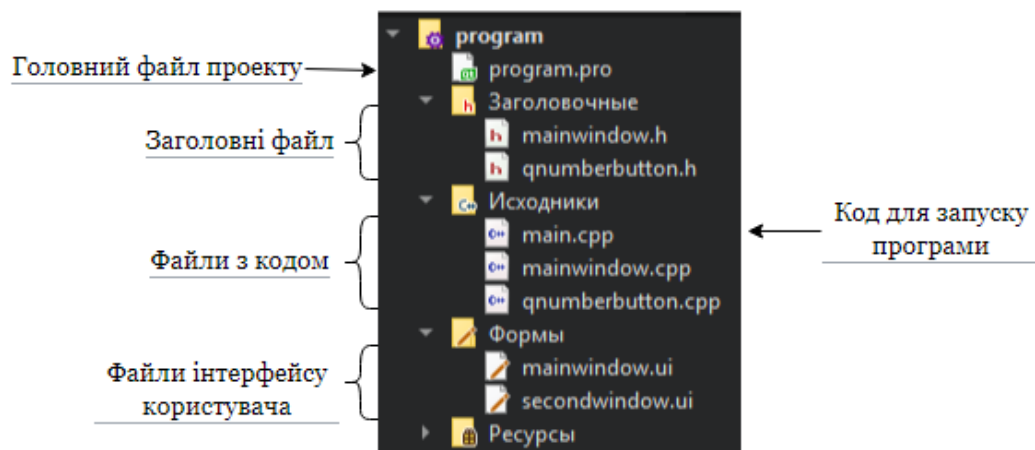


Рис. 3.1. Файлова структура проекту

Файли, які складають основу системи:

- program.pro містить налаштування всього проекту, також в цьому файлі вказані всі файли, з яких складається проект;
- заголовні файли з розширенням .h слугують для описання структури класів
- файли з розширенням .cpp містять в собі підключений .h файл і опис функцій, описаних в заголовних файлах;
- файл з розширенням .ui файли в яких налаштовуються сцени користувацького інтерфейсу;
- окремо знаходиться файл main.cpp, який автоматично генерується середовищем і слугує для запуску додатку.

Кафедра КІТ (47)				НАУ 21 8 88 000 ПЗ			
Виконав	Гуденко Є.І.			Розробка програми	Літера	Аркуш	Аркушів
Керівник	Моденов Ю.Б.					35	10
Консульт.					411 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

3.1. Створення бази даних

База даних створювалась в програмі DB Browser for SQLite, яка надає можливість створення і заповнення таблиць через інтерфейс, а також за допомогою команд мови SQL.

Таблиця «Номери» яка містить перелік всіх номерів готелів та їх характеристики.

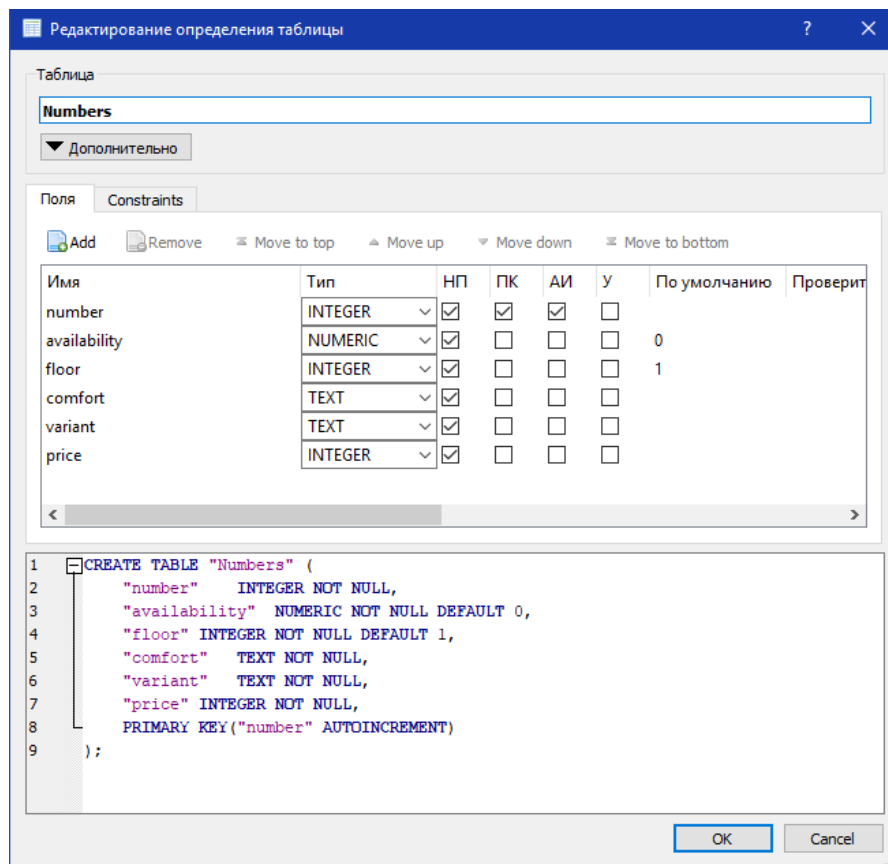


Рис. 3.2. Таблица номерів

Таблиця «Бронювання» яка містить id, який слугує для визначення номеру бронювання, поле номер, яке являється зовнішнім ключем до таблиці номерів, і текстове поле яке буде містити проміжок часу бронювання у вигляді «ддммррррр-ддммррррр»

Редигирование определения таблицы

Таблица: **Booking**

Дополнительно

Поля Constraints

Add Remove Move to top Move up Move down Move to bottom

Имя	Тип	НП	ПК	АИ	У	По умолчанию	Проверит
id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
number	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
timing	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Booking" (
2     "id" INTEGER NOT NULL UNIQUE,
3     "number" INTEGER NOT NULL,
4     "timing" TEXT NOT NULL,
5     PRIMARY KEY("id" AUTOINCREMENT)
6 );

```

OK Cancel

Рис. 3.3. Таблица бронювань

Таблица «Клієнтів» містить поле ПІБ, яке слугує первинним ключем, поле з паспортними даними, поле рейтингу, і поле з коментарем, яке може містити інформацію про номер і час заселення, і також сюди додається інформація яку можна записати при бронюванні номеру.

Редигирование определения таблицы

Таблица: **Clients**

Дополнительно

Поля Constraints

Add Remove Move to top Move up Move down Move to bottom

Имя	Тип	НП	ПК	АИ	У	По умолчанию	Проверит
PIB	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
passport	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
rating	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
comment	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Clients" (
2     "PIB" TEXT NOT NULL UNIQUE,
3     "passport" TEXT NOT NULL UNIQUE,
4     "rating" INTEGER,
5     "comment" TEXT,
6     PRIMARY KEY("PIB")
7 );

```

OK Cancel

Рис. 3.4. Таблица клієнтів

3.2. Створення логіки додатку

Програму умовно можна розділити на 3 частини:

- система завантаження і запису даних до БД;
- система для взаємодії з інтерфейсом;
- система управління.

Кожна частина відповідає за свій функціонал, має різні методи, працює з різними даними, але той же час об'єднані в одну систему.

3.2.1. Створення системи завантаження і запису даних до БД;

Для мінімізації використання ресурсів система клієнт-сервер-бд була спрощена до формату клієнт-бд. Так, як додаток створюється для малого підприємства під його потреби це не викличе проблем.

Програмна реалізація завантаження даних з БД зображена на рис. 3.5.

```
QSqlDatabase db;  
db = QSqlDatabase::addDatabase("QSQLITE");  
db.setDatabaseName("D:\\Project\\database.db");  
db.open();  
  
if (!db.open())  
    QMessageBox::warning(this, "Error", "Error");  
  
QSqlQuery query;  
query.exec("SELECT * FROM Numbers");
```

Рис. 3.5. Програмна реалізація завантаження даних з БД

Програма має завантажувати інформацію про бронювання номера. Для реалізації використовується запит мовою SQL. На рис. 3.3. можна побачити, що таблиця має три поля, а програма заповнює тільки два, бо поле id має автоматичний AI.

```

QSqlQuery query;
query.exec("INSERT INTO Booking (number, timing) " "VALUES (:number, :timing)");
query.bindValue(":number", this->getNumber());
query.bindValue(":timing", firstDate + "-" + secondDate);

```

Рис. 3.6. Програмна реалізація завантаження даних в БД

3.2.2. Створення системи для взаємодії з інтерфейсом

Головний інтерфейс реалізується у файлу `mainwindow.ui` за допомогою Qt Designer, який дає можливість гнучко налаштувати вікно і віджети.

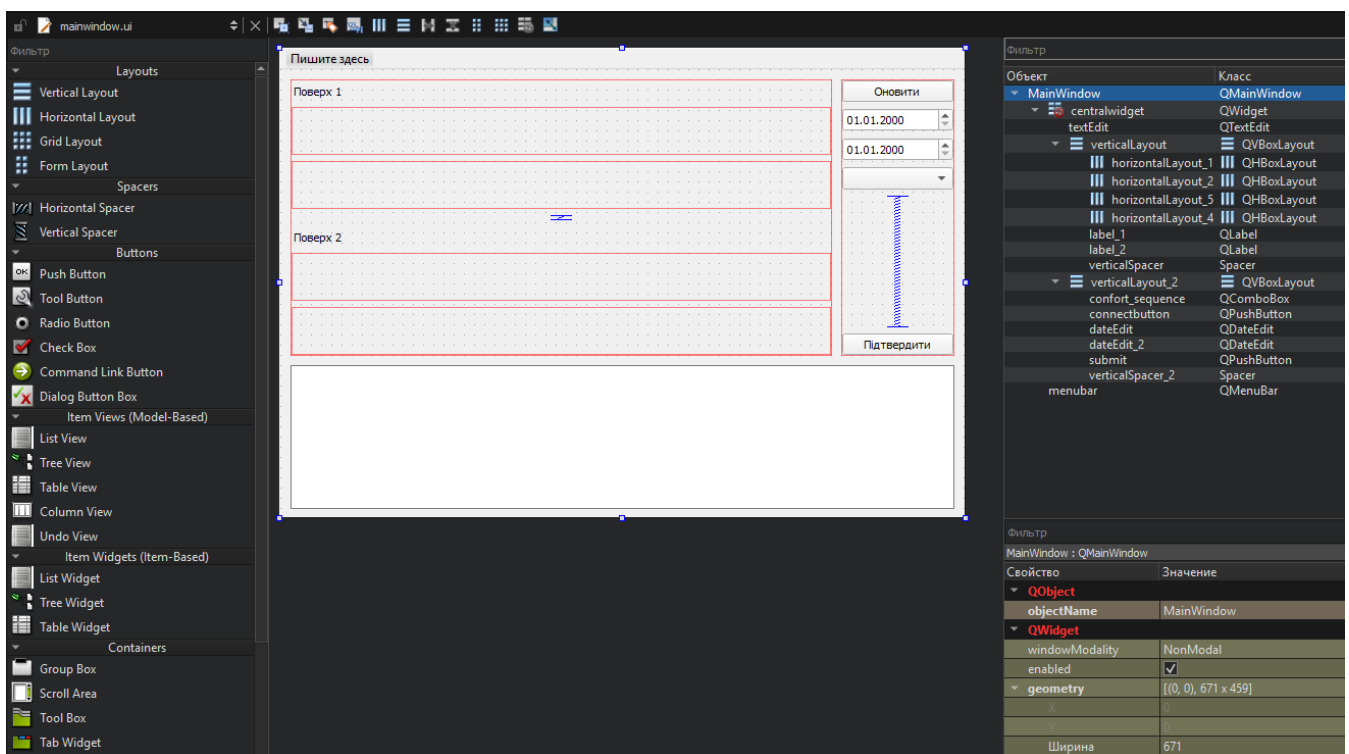


Рис. 3.7. Інтерфейс Qt Designer з файлом головного вікна

В лівій частині інтерфейсу Qt Designer можна побачити список доступних віджетів. В центральній частині знаходиться попередній перегляд головного вікна програми. Справа знаходиться структура вкладеності об'єктів і меню налаштування вибраного віджета.

Для реалізації вікна використовувалися такі віджети:

- QLabel використовується для відображення тексту або зображення. У нього не передбачено взаємодію з користувачем.
- QComboBox використовується для вибору, який відображає поточний елемент і може відображати список, що випадає обраних елементів. Список при цьому може бути редагований, дозволяючи користувачеві змінювати кожен елемент в даному списку.
- QSpinBox використовується для обробки цілих чисел і дискретних наборів значень. Даний віджет дозволяє користувачеві вказати потрібне значення або вручну, або клікаючи мишкою по кнопках вгору або вниз, або натискаючи клавіші клавіатури вгору або вниз для збільшення або зменшення значення, яке відображається в поточний момент. В програмі використовується для вибору дати.
- QLineEdit або QTextEdit використовується як редактор тексту. Редактор рядка дозволяє користувачеві вводити звичайний текст і при цьому використовувати такі функції редагування, як: скасування, повтор, вирізання, вставка, а також перетягування за допомогою механізму drag-and-drop.

Так, як кількість номерів величина змінна, їх генерація в інтерфейсі динамічна.

Для чого був створений окремий клас, про що пізніше. Після програмного створення кнопки номера, її необхідно розмістити на головному вікні, для чого використовується наступний код.


```

QString number = query.value(0).toString();
QString availability = query.value(1).toString();
QString comfort = query.value(3).toString();
QString variant = query.value(4).toString();
QString price = query.value(5).toString();

QNumberButton *t_button = new QNumberButton(this, number, availability, comfort, variant, price);
t_button->setText(t_button->getNumber());
t_button->setFixedSize(QSize(40, 40));
nbutton_sequence.push_back(t_button);

ui->label_1->setEnabled(1);
ui->label_2->setEnabled(1);

if(line < 10)
    vl_sequence[0]->addWidget(t_button);
else if(line < 20)
    vl_sequence[1]->addWidget(t_button);
else if(line < 30)
    vl_sequence[2]->addWidget(t_button);
else if(line < 40)
    vl_sequence[3]->addWidget(t_button);
connect(t_button, SIGNAL(clicked()), this, SLOT(slotGetNumber()));

```

Рис. 3.8. Реалізація розміщення QNumberButton

Код розроблений універсально під будь-яку кількість номерів, що дозволяє не змінювати програму при змінах на підприємстві.



Рис. 3.9. Вид номерів після автоматичного розміщення

Також в програмі реалізоване друге вікно, яке дозволяє наглядно оцінити наповненість готелю(рис. 3.10.). Програмно реалізується через доступ до БД, а саме до таблиці «Booking», з якої береться інформація про період бронювання. Після розшифрування інформація заноситься в таблицю за допомогою кольорів. Зеленим кольором позначаються номери в яких проживають клієнти, фіолетовим майбутні

бронювання. При натисканні на кольоровий квадрат з'являється вікно (рис. 3.11.) з інформацією про клієнта, яка також береться з БД.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
27.05.21		Green		Green		Green	Green		Green	Green			Green				Green		Green	
28.05.21		Green		Green			Green			Green		Purple					Green		Green	
29.05.21		Purple		Purple		Purple	Purple					Purple					Green		Green	
30.05.21		Purple		Purple		Purple	Purple					Purple							Purple	
31.05.21		Purple		Purple		Purple	Purple		Purple			Purple							Purple	
01.06.21				Purple					Purple										Purple	
02.06.21				Purple					Purple				Purple						Purple	
03.06.21				Purple					Purple				Purple							
04.06.21				Purple					Purple				Purple			Purple				
05.06.21				Purple									Purple				Purple			
06.06.21																	Purple			
07.06.21																	Purple			
08.06.21																	Purple			
09.06.21																				

Рис. 3.10. Вікно наповненості готелю

Dialog
 ПІБ: Xxxxxx Xxxxxx Xxxxxx
 Паспортні дані: xxxxxxxxxxxxxxxxxxxx
 Дата відвідування: 29.05.21 - 31.05.21
 Рейтинг: 0

Рис. 3.11. Вікно інформації про клієнта

3.2.3. Створення системи управління

Управління у головному вікні програми відбувається кнопками номерів, які реалізовані через клас `QNumberButton` (Додаток В). Кнопки містять всередині себе всі данні, що характеризують номер. При виборі необхідних параметрів номеру на

головному вікні, номери які підходять під вибрані характеристики виділяються різними кольорами [15].

```
if(comfort == ui->comfort_sequence->currentText())
    if(data == nbutton_sequence.at(i)->getAvailability())
        nbutton_sequence.at(i)->setStyleSheet("background-color: #83f7aa"); // green
    else nbutton_sequence.at(i)->setStyleSheet("background-color: #9383f7"); // purple
else nbutton_sequence.at(i)->setStyleSheet("background-color: #fff"); //white
```

Рис. 3.12. Реалізація алгоритму оптимального розміщення

3.3. Демонстрація роботи програми

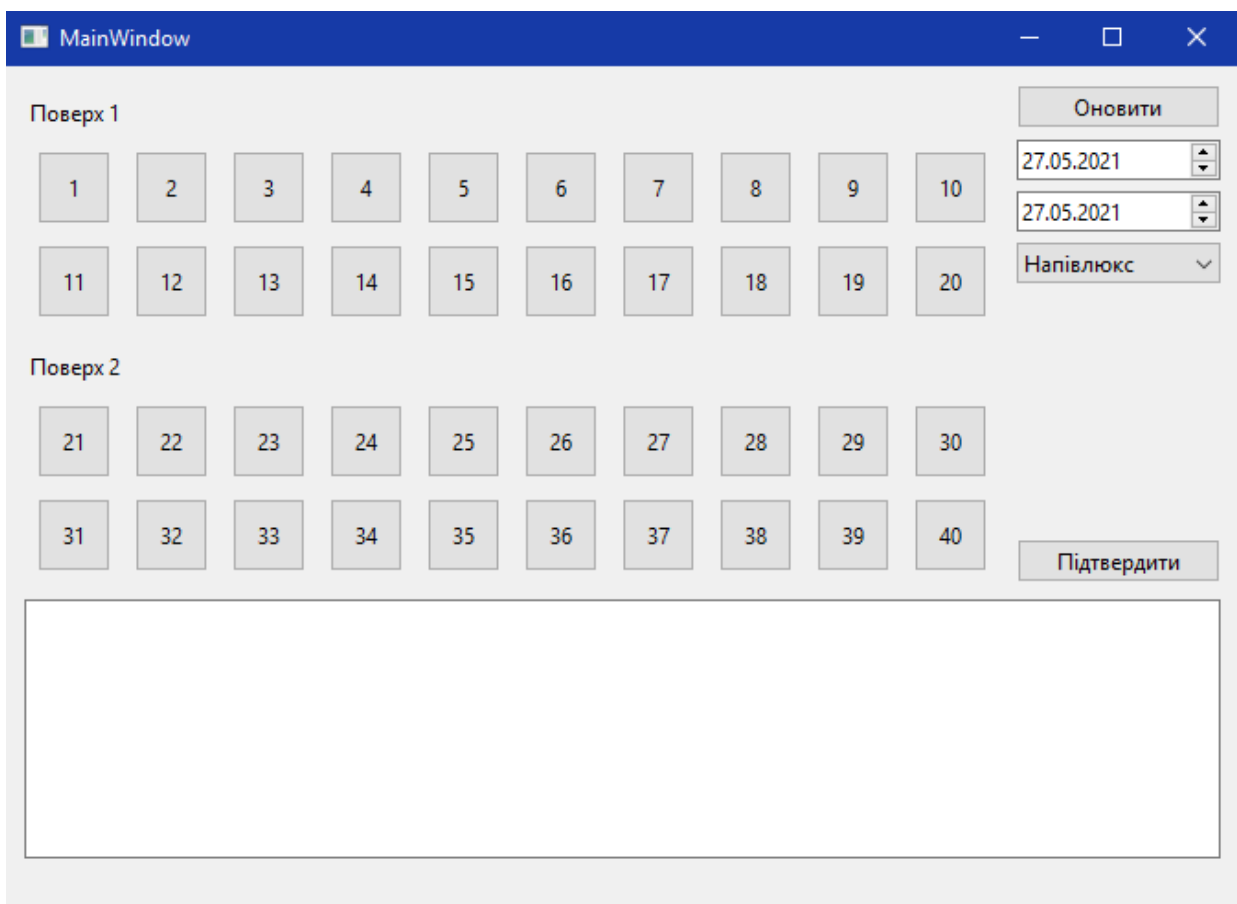


Рис. 3.13. Головне вікно програми після запуску

За допомогою меню в правій частині вікна можна вибрати термін бронювання, а також комфортність номеру. Після натискання кнопки оновити, номери змінять свій колір відносно до описаного вище алгоритму.

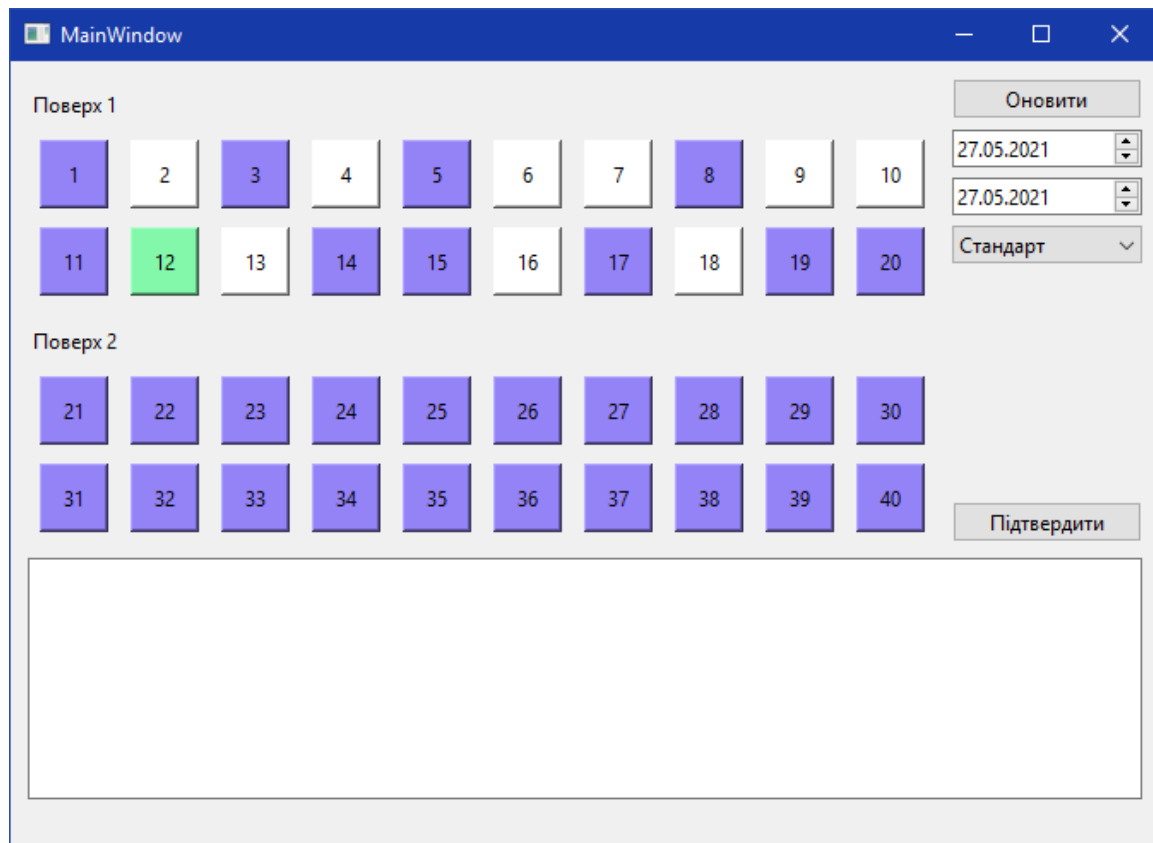


Рис. 3.14. Головне вікно програми після вибору параметрів

Можна побачити, що кнопки зміни колір відповідно до порядку, в якому необхідно перевірити номер. Тобто якщо кнопка стала білою, тоді номер зайнятий, або не підходить під вибрані параметри, фіолетовий колір говорить про те, що номер вільний і підходить під вибрані умови, але при заселенні в даний номер з'являється проміжок між клієнтами, що може значно повпливати на економічну складову готелю. Зеленим кольором відмічаються лише ті номери, які відповідають всім критеріям, а також дата заселення наступного клієнта знаходиться в той самий день, коли з'їзжають попередні, що дозволяє максимально використовувати місткість. Якщо подивитися на рис. 3.10 на ньому видно, що номер 12 ідеально підходить під вибрані критерії, що 27.05 відбувається заселення і 28.05 зранку виселення, після чого в номері проводиться прибирання і 28.05 відбувається заселення нових клієнтів.

Після натискання на кнопку з номером 12 в інформаційному полі внизу отримаємо інформацію відповідно вибраного номера.

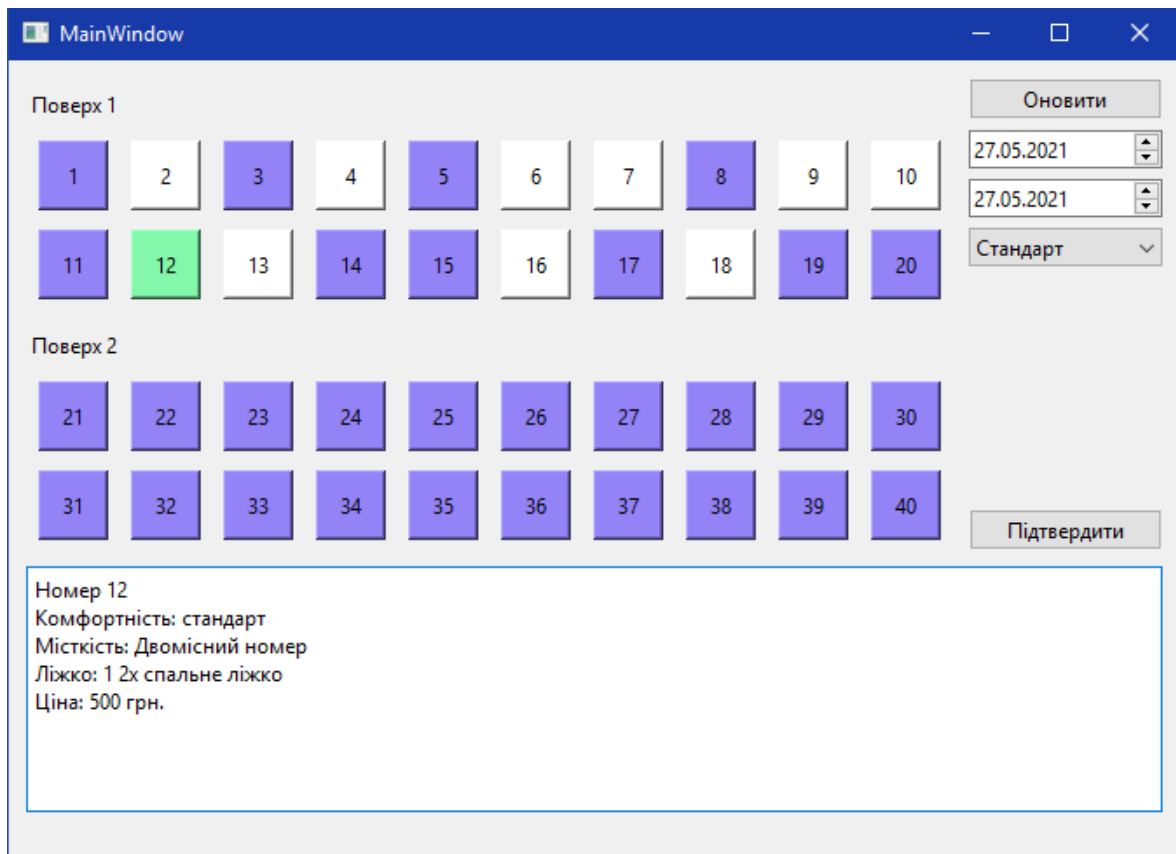


Рис. 3.15. Головне вікно програми після вибору номера

Після чого натискається кнопка підтвердити, і з'являється вікно для введення персональної інформації клієнта. Після введення ПІБ даних, виконається перевірка на збіг даних з попередніми відвідувачами. Якщо збіг відбудеться з'явиться вікно типу рис. 3.11., якщо ні то після заповнення паспортних даних і закриття вікна дані будуть внесені в БД в таблицю клієнти.

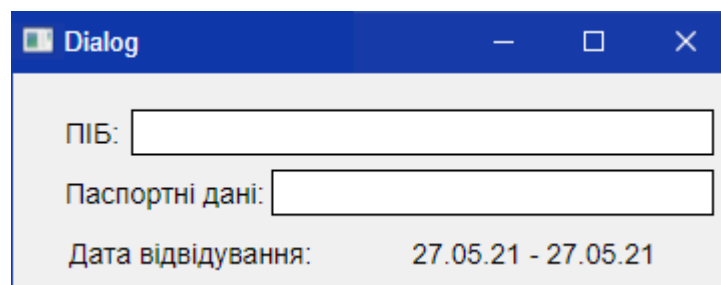


Рис. 3.16. Вікно для внесення інформації про клієнта

Для того щоб отримати інформацію, що до наповненості готелю рис. 3.10. використовується введення «0» в текстову область і натискання кнопки підтвердити.

В цьому вікні відображається 2 неділі (14 днів) від вказаної зверху дати, інформації про наповненість готелю. На листі розміщається 20 номерів, тому є перемикач поверхів, для наступних 20 – 40 номерів.

Висновки до розділу 3

Основними етапами розробки системи є створення бази даних і налаштування доступу до неї, створення класу який дозволить обробляти інформацію і демонструвати результати в графічному інтерфейсі.

В результаті виконання описаних етапів була створена програма яка дозволить спростити процедуру бронювання і надає можливість наглядного перегляду наповненості готелю, якою можуть користуватися співробітники організації за допомогою розробленого інтерфейсу.

Створена програма має оптимізувати витрати часу, що в свою чергу зменшить економічні. Під час проектування і написання було враховано всі вимоги відповідно до поставленої задачі.

ВИСНОВКИ

На сьогодні більшість організацій намагається впровадити в робочий процес новітні технології для оптимізації робочого процесу співробітників, що призводить до збільшення конкурентної спроможності компанії та підвищення продуктивності роботи.

В даному проекті була розроблена автоматизована система управління готелем. Її актуальність перш за все зумовлена необхідністю автоматизації процесів бронювання номерів та заселення гостей.

В проекті були проаналізовані наявні аналоги АСУ готелями, і також зроблено висновки, що їх використання в даній організації є цілком недоречне. У проекті була розроблена функціональна схема системи та схема БД.

Програмний продукт створено з застосуванням сучасних засобів програмування таких, як Qt Creator для написання функціональної частини і DB Browser for SQLite для БД. Розроблена база даних задовольняє поставленим вимогам до створеної системи.

Створену програму було досконало протестовано. Проект в цілому відповідає поставленим вимогам і відповідає завданню. Після завершення програмування і тестування продукту можна сказати, що версія системи керування готелем повністю завершена. В майбутньому можна додати всі можливі елементи, як в проаналізованих АСУ, тоді програма буде повністю відповідати і забезпечувати весь можливий функціонал.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пуцентейло П.Р. Економіка і організація туристично-готельного підприємництва [Електронний ресурс]. Режим доступу: <https://tourlib.net> (дата звернення 15.05.2021). – Назва з екрана.
2. Какие есть типы номеров в отелях? [Електронний ресурс]. Режим доступу: <https://hotellook.ru> (дата звернення 15.05.2021). – Назва з екрана.
3. Менеджмент туризму [Електронний ресурс]. Режим доступу: <https://library.if.ua> (дата звернення 15.05.2021). – Назва з екрана.
4. Рябенюк М.О., Троян В.В. Застосування новітніх інформаційних систем управління готелем [Електронний ресурс]. Режим доступу: <https://tourlib.net> (дата звернення 15.05.2021). – Назва з екрана.
5. Программа для гостиницы АСУ «Эдельвейс» [Електронний ресурс]. Режим доступу: <https://biznesplan-primer.ru> (дата звернення 15.05.2021). – Назва з екрана.
6. Про хмарну систему jSolutions [Електронний ресурс]. Режим доступу: <https://jsolutions.ua> (дата звернення 15.05.2021). – Назва з екрана.
7. Мануал «Fidelio V8» [Електронний ресурс]. Режим доступу: <https://umhos.ru> (дата звернення 15.05.2021). – Назва з екрана.
8. Перспективи впровадження автоматизованих систем управління в готелях України [Електронний ресурс]. Режим доступу: <http://kafmen.ru> (дата звернення 15.05.2021). – Назва з екрана.
9. RDDMS [Електронний ресурс]. Режим доступу: <https://ru.bmstu.wiki> (дата звернення 15.05.2021). – Назва з екрана.
10. Профессиональное программирование на C++ [Електронний ресурс]. Режим доступу: <https://codernet.ru> (дата звернення 15.05.2021). – Назва з екрана.
11. Виджеты в Qt5 [Електронний ресурс]. Режим доступу: <https://ravesli.com> (дата звернення 15.05.2021). – Назва з екрана.

12. Microsoft Visual Studio [Электронный ресурс]. Режим доступа: <https://uk.wikipedia.org> (дата звернення 15.05.2021). – Назва з екрана.
13. Visual Studio Code [Электронный ресурс]. Режим доступа: <https://uk.wikipedia.org> (дата звернення 15.05.2021). – Назва з екрана.
14. MonoDevelop [Электронный ресурс]. Режим доступа: <https://uk.wikipedia.org> (дата звернення 15.05.2021). – Назва з екрана.
15. Динамическое создание виджетов в Qt [Электронный ресурс]. Режим доступа: <https://evileg.com> (дата звернення 15.05.2021). – Назва з екрана.

ДОДАТКИ

Додаток А

Реалізація головного файлу компілятора program.pro

```
QT      += core gui sql

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs
deprecated before Qt 6.0.0

SOURCES += \
    main.cpp \
    mainwindow.cpp \
    qnumberbutton.cpp

HEADERS += \
    mainwindow.h \
    qnumberbutton.h

FORMS += \
    mainwindow.ui \
    secondwindow.ui

# Default rules for deployment.
qnx: target.path = /tmp/${TARGET}/bin
else: unix:!android: target.path = /opt/${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

RESOURCES += \
    resource.qrc
```

Програмна реалізація класу MainWindow

mainwindow.h:

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <qnumberbutton.h>
#include <QString>
#include <QList>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
    int getNumber();

private slots:
    void widgen_update();

    void on_connectbutton_clicked();
    void slotGetNumber();

    void on_dateEdit_userDateChanged(const QDate &date);

    void on_dateEdit_2_userDateChanged(const QDate &date);

    void on_confort_sequence_currentIndexChanged(int index);

    void on_submit_clicked();

private:
    Ui::MainWindow *ui;
    int i_number[5] = {1, 2, 3, 4, 5};
    QString str_names[5] = {"n1", "n2", "n3", "n4", "n5"};
    QList <QNumberButton*> nbutton_sequence;
};
#endif // MAINWINDOW_H

```

mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QtSql/QtSql>
#include <QtSql/QtSqlDatabase>
#include <QtSql/QtSqlQuery>

```

```

#include <QString>
#include <QMessageBox>
#include <QPixmap>
#include <QDateEdit>
#include <anotherwindow.h>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    ui->dateEdit->setDate(QDate::currentDate());
    ui->dateEdit_2->setDate(QDate::currentDate());
    QList<QString> list_comfort= {"Напівлюкс", "Люкс", "Стандарт", "Сімейний",
    "Апартаменти"};
    ui->confort_sequence->addItem(list_comfort);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::widgen_update()
{
    for(int i = 0; i < nbutton_sequence.size(); ++i)
    {
        QString comfort = "", data = "";

        if(comfort == ui->confort_sequence->currentText())
            if(data == nbutton_sequence.at(i)->getAvailability())
                nbutton_sequence.at(i)->setStyleSheet("background-color:
#83f7aa"); // green
            else nbutton_sequence.at(i)->setStyleSheet("background-color:
#9383f7"); // purple
            else nbutton_sequence.at(i)->setStyleSheet("background-color: #fff");
//white
    }
}

void MainWindow::on_connectbutton_clicked()
{
    QSqlDatabase db;
    db = QSqlDatabase::addDatabase("SQLITE");
    db.setDatabaseName("D:\\Project\\database.db");
    db.open();

    if (!db.open())
        QMessageBox::warning(this, "Error", "Error");

    QSqlQuery query;
    query.exec("SELECT * FROM Numbers");

    QHBoxLayout* vl_sequence[4] = {ui->horizontalLayout_1, ui->horizontalLayout_2,
    ui->horizontalLayout_5, ui->horizontalLayout_4};

    for(int line = 0; query.next(); ++line)
    {
        QString number = query.value(0).toString();
    }
}

```

```

    QString availability = query.value(1).toString();
    QString comfort = query.value(3).toString();
    QString variant = query.value(4).toString();
    QString price = query.value(5).toString();

    QNumberButton *t_button = new QNumberButton(this, number, availability,
comfort, variant, price);
    t_button->setText(t_button->getNumber());
    t_button->setFixedSize(QSize(40, 40));
    nbutton_sequence.push_back(t_button);

    ui->label_1->setEnabled(1);
    ui->label_2->setEnabled(1);

    if(line < 10)
        vl_sequence[0]->addWidget(t_button);
    else if(line < 20)
        vl_sequence[1]->addWidget(t_button);
    else if(line < 30)
        vl_sequence[2]->addWidget(t_button);
    else if(line < 40)
        vl_sequence[3]->addWidget(t_button);
    connect(t_button, SIGNAL(clicked()), this, SLOT(slotGetNumber()));
}
db.close();

}

void MainWindow::slotGetNumber()
{
    QNumberButton *button = (QNumberButton*) sender();
    ui->textEdit->setText(button->getPrice());
}

void MainWindow::on_dateEdit_userDateChanged(const QDate &date)
{
    if(date < QDate::currentDate())
        ui->dateEdit->setDate(QDate::currentDate());
}

void MainWindow::on_dateEdit_2_userDateChanged(const QDate &date)
{
    if(date < ui->dateEdit->date())
        ui->dateEdit_2->setDate(ui->dateEdit->date());
}

void MainWindow::on_comfort_sequence_currentIndexChanged(int index)
{
    //ui->textEdit->setText(QString::number(index));
    //ui->textEdit->setText(QDate::currentDate().toString("ddMMyyyy"));
    widgen_update();
}

void MainWindow::on_submit_clicked()
{
}

```

Програмна реалізація класу QNumberButton

qnumberbutton.h:

```
#ifndef QNUMBERBUTTON_H
#define QNUMBERBUTTON_H

#include <QPushButton>
#include <QString>

class QNumberButton : public QPushButton
{
    Q_OBJECT
public:
    explicit QNumberButton(QWidget *parent = 0,   QString number = "", QString
availability   = "", QString comfort = "", QString variant = "", QString price =
"");
    ~QNumberButton();
    static int ResID;
    int getID();
    QString getNumber();
    QString getPrice();
    QString getAvailability();

private:
    int buttonID = 0;
    QString number;
    QString availability;
    QString comfort;
    QString variant;
    QString price;
};
#endif // QNUMBERBUTTON_H
```

qnumberbutton.cpp:

```
#include "qnumberbutton.h"

QNumberButton::QNumberButton(QWidget *parent,   QString number, QString
availability, QString comfort, QString variant, QString price)
    : QPushButton(parent), number(number), availability(availability),
comfort(comfort), variant(variant), price(price)
{
    ResID++;
    buttonID = ResID;
}

QNumberButton::~QNumberButton() {}

int QNumberButton::getID()
{
    return buttonID;
}

int QNumberButton::ResID = 0;
```

```
QString QNumberButton::getNumber()  
{  
    return number;  
}  
  
QString QNumberButton::getPrice()  
{  
    return price;  
}  
  
QString QNumberButton::getAvailability()  
{  
    return availability;  
}
```

Програмна реалізація класу main.cpp

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```